Uebungsblatt 5
„Mustererkennung“

J. Cavojska, N. Lehmann, R. Toudic

01.06.2015

Inhaltsverzeichnis

1 Schnitte von zwei Gaußkurven

$$f_1(x, \mu_1, \sigma_1) = \frac{1}{\sqrt{2\pi\sigma_1^2}} \cdot e^{-\frac{(x-\mu_1)^2}{2\sigma_1^2}}$$
$$f_2(x, \mu_2, \sigma_2) = \frac{1}{\sqrt{2\pi\sigma_2^2}} \cdot e^{-\frac{(x-\mu_2)^2}{2\sigma_2^2}}$$

Eigenschaften von Dichtfunktionen f_i für $\sigma_i > 0$ mit $i \in \mathbb{N}$:

- f_i ist achsensymmeterisch um μ_i
- f_i hat zwei Wendepunkte: $\mu_i - \sigma_i$ und $\mu_i + \sigma_i$
- f_i hat genau ein Maximum bei μ_i
- f_i ist stetig / für jede reelle Zahl definiert
- $f_i(x, \mu_i, \sigma_i) > 0$

Eigenschaften von Dichtfunktionen f_i für $\sigma_i = 0$ mit $i \in \mathbb{N}$:

- f_i ist nicht definiert
- philosophische Betrachtung: f_i ist eine Konstante!?

Eigenschaften von Dichtfunktionen f_i für $\sigma_i < 0$ mit $i \in \mathbb{N}$:

- f_i ist nicht definiert in $\mathbb{R}$ (, aber in $\mathbb{C}$)

Schnittpunkte von f_1 und f_2 bestimmen durch Gleichsetzung:

$$
\begin{aligned}
f_1(x,\mu_1,\sigma_1) &= f_2(x,\mu_2,\sigma_2) && \Leftrightarrow \\
\frac{1}{\sqrt{2\pi\sigma_1^2}} \cdot e^{-\frac{(x-\mu_1)^2}{2\sigma_1^2}} &= \frac{1}{\sqrt{2\pi\sigma_2^2}} \cdot e^{-\frac{(x-\mu_2)^2}{2\sigma_2^2}} && \Leftrightarrow \\
\frac{\sqrt{2\pi\sigma_2^2}}{\sqrt{2\pi\sigma_1^2}} \cdot e^{-\frac{(x-\mu_1)^2}{2\sigma_1^2}} &= e^{-\frac{(x-\mu_2)^2}{2\sigma_2^2}} && \Leftrightarrow \\
\frac{\sqrt{2\pi\sigma_2^2}}{\sqrt{2\pi\sigma_1^2}} &= e^{-\frac{(x-\mu_2)^2}{2\sigma_2^2}+\frac{(x-\mu_1)^2}{2\sigma_1^2}} && \Leftrightarrow \\
\frac{\sqrt{\sigma_2^2}}{\sqrt{\sigma_1^2}} &= e^{-\frac{(x-\mu_2)^2}{2\sigma_2^2}+\frac{(x-\mu_1)^2}{2\sigma_1^2}} && \Leftrightarrow \\
ln\left(\frac{\sqrt{\sigma_2^2}}{\sqrt{\sigma_1^2}}\right) &= -\frac{(x-\mu_2)^2}{2\sigma_2^2} + \frac{(x-\mu_1)^2}{2\sigma_1^2} && \Leftrightarrow \\
2\sigma_2^2 \cdot ln\left(\frac{\sqrt{\sigma_2^2}}{\sqrt{\sigma_1^2}}\right) &= -(x-\mu_2)^2 + \frac{2\sigma_2^2(x-\mu_1)^2}{2\sigma_1^2} && \Leftrightarrow \\
2\sigma_2^2 \cdot ln\left(\frac{\sqrt{\sigma_2^2}}{\sqrt{\sigma_1^2}}\right) &= -(x-\mu_2)^2 + \frac{\sigma_2^2(x-\mu_1)^2}{\sigma_1^2} && \Leftrightarrow \\
2\sigma_1^2\sigma_2^2 \cdot ln\left(\frac{\sqrt{\sigma_2^2}}{\sqrt{\sigma_1^2}}\right) &= -\sigma_1^2(x-\mu_2)^2 + \sigma_2^2(x-\mu_1)^2 && \Leftrightarrow \\
2\sigma_1^2\sigma_2^2 \cdot ln\left(\frac{\sqrt{\sigma_2^2}}{\sqrt{\sigma_1^2}}\right) &= -\sigma_1^2(x^2-2x\mu_2+\mu_2^2) + \sigma_2^2(x-\mu_1)^2 && \Leftrightarrow \\
2\sigma_1^2\sigma_2^2 \cdot ln\left(\frac{\sqrt{\sigma_2^2}}{\sqrt{\sigma_1^2}}\right) &= -\sigma_1^2(x^2-2x\mu_2+\mu_2^2) + \sigma_2^2(x^2-2x\mu_1+\mu_1^2) && \Leftrightarrow \\
2\sigma_1^2\sigma_2^2 \cdot ln\left(\frac{\sqrt{\sigma_2^2}}{\sqrt{\sigma_1^2}}\right) &= -\sigma_1^2x^2 + 2x\mu_2\sigma_1^2 - \mu_2^2\sigma_1^2 + \sigma_2^2(x^2-2x\mu_1+\mu_1^2) && \Leftrightarrow \\
2\sigma_1^2\sigma_2^2 \cdot ln\left(\frac{\sqrt{\sigma_2^2}}{\sqrt{\sigma_1^2}}\right) &= -\sigma_1^2x^2 + 2x\mu_2\sigma_1^2 - \mu_2^2\sigma_1^2 + \sigma_2^2x^2 - 2x\mu_1\sigma_2^2 + \mu_1^2\sigma_2^2 && \Leftrightarrow \\
2\sigma_1^2\sigma_2^2 \cdot ln\left(\frac{\sqrt{\sigma_2^2}}{\sqrt{\sigma_1^2}}\right) + \mu_2^2\sigma_1^2 &= -\sigma_1^2x^2 + 2x\mu_2\sigma_1^2 + \sigma_2^2x^2 - 2x\mu_1\sigma_2^2 + \mu_1^2\sigma_2^2 && \Leftrightarrow \\
2\sigma_1^2\sigma_2^2 \cdot ln\left(\frac{\sqrt{\sigma_2^2}}{\sqrt{\sigma_1^2}}\right) + \mu_2^2\sigma_1^2 - \mu_1^2\sigma_2^2 &= \underset{3}{-}\sigma_1^2x^2 + 2x\mu_2\sigma_1^2 + \sigma_2^2x^2 - 2x\mu_1\sigma_2^2 && \Leftrightarrow \\
2\sigma_1^2\sigma_2^2 \cdot ln\left(\frac{\sqrt{\sigma_2^2}}{\sqrt{\sigma_1^2}}\right) + \mu_2^2\sigma_1^2 - \mu_1^2\sigma_2^2 &= x^2(\sigma_2^2-\sigma_1^2) + 2x\mu_2\sigma_1^2 - 2x\mu_1\sigma_2^2 && \Leftrightarrow
\end{aligned}
$$

$$2\sigma_1^2\sigma_2^2 \cdot ln\left(\frac{\sqrt{\sigma_2^2}}{\sqrt{\sigma_1^2}}\right) + \mu_2^2\sigma_1^2 - \mu_1^2\sigma_2^2 = x^2(\sigma_2^2 - \sigma_1^2) + 2x(\mu_2\sigma_1^2 - \mu_1\sigma_2^2) \quad \Leftrightarrow$$

$$\frac{2\sigma_1^2\sigma_2^2 \cdot ln\left(\frac{\sqrt{\sigma_2^2}}{\sqrt{\sigma_1^2}}\right) + \mu_2^2\sigma_1^2 - \mu_1^2\sigma_2^2}{\sigma_2^2 - \sigma_1^2} = x^2 + \frac{2x(\mu_2\sigma_1^2 - \mu_1\sigma_2^2)}{\sigma_2^2 - \sigma_1^2}$$

Ab hier gibt es zwei Möglichkeiten, die Gleichung zu lösen...

1. quadratische Ergänzung mit dem Term $\left(\frac{2(\mu_2\sigma_1^2 - \mu_1\sigma_2^2)}{\sigma_2^2 - \sigma_1^2}\right)^2$ und umformen nach x
2. Term umformen und die P-Q-Formel verwenden

Wir haben uns für die 2. Option entschieden:

$$x^2 + \frac{2x(\mu_2\sigma_1^2 - \mu_1\sigma_2^2)}{\sigma_2^2 - \sigma_1^2} - \frac{2\sigma_1^2\sigma_2^2 \cdot ln\left(\frac{\sqrt{\sigma_2^2}}{\sqrt{\sigma_1^2}}\right) + \mu_2^2\sigma_1^2 - \mu_1^2\sigma_2^2}{\sigma_2^2 - \sigma_1^2} = 0 \qquad \Leftrightarrow$$

$$x^2 + x \cdot \underbrace{\frac{2\mu_2\sigma_1^2 - 2\mu_1\sigma_2^2}{\sigma_2^2 - \sigma_1^2}}_{P} - \underbrace{\frac{2\sigma_1^2\sigma_2^2 \cdot ln\left(\frac{\sigma_2}{\sigma_1}\right) + \mu_2^2\sigma_1^2 - \mu_1^2\sigma_2^2}{\sigma_2^2 - \sigma_1^2}}_{Q}$$

Nun können wir die quadratische Gleichung mit der P-Q-Formel lösen.
Zur Erinnerung: Die P-Q-Formel lautet: $x_{1/2} = -\frac{P}{2} \pm \sqrt{\left(\frac{P}{2}\right)^2 - Q}$.

$$x_1 = 4 \cdot \frac{\mu_1\sigma_2^2 - \mu_2\sigma_1^2}{\sigma_1^2 - \sigma_2^2} + \sqrt{\left(\frac{2\mu_2\sigma_1^2 - 2\mu_1\sigma_2^2}{\sigma_2^2 - \sigma_1^2}\right)^2 + \frac{2\sigma_1^2\sigma_2^2 \cdot ln\left(\frac{\sigma_2}{\sigma_1}\right) + \mu_2^2\sigma_1^2 - \mu_1^2\sigma_2^2}{\sigma_2^2 - \sigma_1^2}}$$

$$x_2 = 4 \cdot \frac{\mu_1\sigma_2^2 - \mu_2\sigma_1^2}{\sigma_1^2 - \sigma_2^2} - \sqrt{\left(\frac{2\mu_2\sigma_1^2 - 2\mu_1\sigma_2^2}{\sigma_2^2 - \sigma_1^2}\right)^2 + \frac{2\sigma_1^2\sigma_2^2 \cdot ln\left(\frac{\sigma_2}{\sigma_1}\right) + \mu_2^2\sigma_1^2 - \mu_1^2\sigma_2^2}{\sigma_2^2 - \sigma_1^2}}$$

Sonderfall: Sei $\sigma = \sigma_1 = \sigma_2$, dann gilt $f_1(x, \sigma, \mu_1) = f_2(x, \sigma, \mu_2) = \frac{\mu_1 + \mu_2}{2}$.

2 Klassifikation mit Fisher-Diskriminante

```
% Clean up
clear all
close all
clc

% Datenaufbreitung
Data         = load('fisher.txt');
Data0        = Data((Data(:,3)==0),:);
Data1        = Data((Data(:,3)==1),:);
Koordinaten  = Data(:,1:2);
Koordinaten0 = Data((Data(:,3)==0),1:2);
Koordinaten1 = Data((Data(:,3)==1),1:2);
Klassen      = Data(:,3);

% Aufgabe 2

% Grafik erstellen
figure('NumberTitle','off','Name','Aufgabe 2 - Bildpunkte');

X = Koordinaten(:,1);
Y = Koordinaten(:,2);
min_x = -250;
max_x =  250;
li = min_x:max_x;

% Punkte plotten
gscatter(X,Y,Klassen,'krb','+x',[],'off');
hold on

% Diskriminante erzeugen
FDK    = fitcdiscr(Koordinaten,Klassen);
Konst  = FDK.Coeffs(1,2).Const;
Linear = FDK.Coeffs(1,2).Linear;
fd = @(x1,x2) Linear(1)*x1 + Linear(2)*x2 + Konst;

% Fisher-Diskriminante plotten
Diskriminante = ezplot(fd, [min_x,max_x]);
Diskriminante.Color = 'b';

% Scatter within berechnen
mean1 = mean(Koordinaten0);
mean2 = mean(Koordinaten1);
S1 = 0;
for i = 1:size(Koordinaten0, 1)
    S1 = S1 + (Koordinaten0(i,:) - mean1)' * (Koordinaten0(i,:) - mean1);
end
S2 = 0;
for i = 1:size(Koordinaten1, 1)
    S2 = S2 + (Koordinaten1(i,:) - mean2)' * (Koordinaten1(i,:) - mean2);
end
S_w = S1 + S2;

```

```
% Vektor w berechnen:
w = inv(S_w) * (mean1 - mean2)';
w_norm = w / norm(w)

% Gerade durch den Vektor w legen und plotten
w_gerade_x = w_norm(1) * li;
w_gerade_y = w_norm(2) * li;
plot(w_gerade_y, w_gerade_x, 'g');

% Daten auf Vektor w_norm projezieren
Koordinaten0_p = [];
for i = 1:size(Koordinaten0, 1)
    Koordinaten0_p = vertcat(Koordinaten0_p, Koordinaten0(i, :) * w_norm);
end
Koordinaten1_p = [];
for i = 1:size(Koordinaten1, 1)
    Koordinaten1_p = vertcat(Koordinaten1_p, Koordinaten1(i, :) * w_norm);
end
Koordinaten_p = vertcat(Koordinaten0_p, Koordinaten1_p);

% pdf der projizierten Daten aus Klasse 1 erzeugen
mean1_p = mean(Koordinaten0_p);
std1_p  = std(Koordinaten0_p);
pdf1_p  = pdf('Normal',li,mean1_p, std1_p);

% pdf der projizierten Daten aus Klasse 2 erzeugen
mean2_p = mean(Koordinaten1_p);
std2_p  = std(Koordinaten1_p);
pdf2_p  = pdf('Normal',li,mean2_p, std2_p);

% Schnittpunkt berechnen
[ispt_x,ispt_y] = intersections(li, pdf1_p, li, pdf2_p, 1);

% w0 berechnen und plotten
w0 = ispt_x * (w_norm')
plot(w0(1),w0(2),'m.','markersize',20);

% Titel, Bezeichner und Legende der Grafk
title('Aufgabe 2 - Plot');
xlabel('X-Koordinaten');
ylabel('Y-Koordinaten');
axis([-50 300 -50 300])
legend('1. Klasse','2. Klasse', 'Diskriminante', 'Gerade durch Vektor w', 'Punkt w_0')
```

2.1 Mehr als 2 Klassen klassifizieren mit Binärbaum

Idee:

Mit Hilfe der Fisher-Diskriminanten bauen wir einen binären Entscheidungsbaum auf. In den Blättern stehen die Klassen. Von der Wurzel bis zu den Blättern werden in den Knoten die jeweiligen Diskriminanten gespeichert.

Algorithmus:

1. wähle zufällige Fisher-Diskriminante und speichere sie in der Wurzel
2. prüfe, ob eine weitere Diskriminante die aktuelle Diskriminante schneidet
   - **nein:**
     - Klasse -1 liegt im linken Blatt
     - Klasse +1 liegt im rechten Blatt
     - GOTO: *Punkt 3*
   - **ja:**
     - speichere die neue Diskriminante mit dem Vermerk -1 der aktuellen Diskriminante im linken Kinderknoten und mit dem Vermerk +1 der aktuellen Diskriminante im rechten Kinderknoten
     - markiere aktuelle Diskriminante als bearbeitet
     - wähle wie bei der Breitensuche den nächsten Kinderknoten aus
     - GOTO: *Punkt 2*
3. Algorithmus terminiert

Klappt der Algorithmus?